space. The current era can be compared to the fifteenth and sixteenth centuries, which were pretty exciting times because of the discovery of the New World.

KC: _The realization that there were other civilizations._

Oliver: Which had long been suspected, and which Columbus found to be the case. He thought he was in the Indies -- which were a locality known to the Europeans -- but instead he found something entirely distinct, the Americas. And the excitement of that discovery completely reversed the comparative stagnation of Europe.

I think that this search, if it can be accomplished, would be as great and as positive a change. In the first place, if you contact one extraterrestrial civilization, you probably will contact a network rather than one, because that civilization may well be ahead of us, whether in years, in experience, or in technological aptitude. At that point we find ourselves a member of a community of intelligent cultures, which would mean that the whole natural history of the galaxy might be at our disposal. We could, for example, find out whether DNA is the chemical of life everywhere or whether there are different forms --

KC: _Something based on silicon?_

Oliver: Well, the silicon-based life is going to be the one we fabricate, I think.

KC: _That's true, too._

Oliver: But not the way we're going. I think there is so much difference between the brain and the computer. Their similarities are dwarfed by their differences. We're just going to have to work with multiple models of intelligence and make them cooperate to the best effect we can.

HELLO, SAILOR!
a concise appreciation
of the Stanford Artificial Intelligence Labs

----

by Les Earnest

SAIL grew out of the Stanford Artificial Intelligence Project, which was started by Prof. John McCarthy when he came from MIT in 1962. He and Prof. Marvin Minsky had co-founded the MIT AI Project in the late 1950s, and McCarthy had developed the LISP programming language there.

McCarthy had perceived the need for interactive computing in

that era when most large computers were used exclusively as batch processors. In 1959 he wrote a memo that proposed general purpose timesharing. Part of the inspiration for this idea was a special-purpose timesharing system called SAGE, the air defense control system that was then being developed at MIT Lincoln Lab (by a bunch of people, including me) using hardware manufactured by IBM.

Working with Ed Fredkin at BBN, McCarthy developed an early timesharing system using a DEC PDP-1 computer. Fernando Corbato concurrently developed another one at MIT. Shortly thereafter, Project MAC was initiated at MIT to develop this idea further. McCarthy was invited to head that project, but chose instead to remain focused on artificial intelligence. He moved to Stanford a short time later.

In 1963 at Stanford, McCarthy began developing the first display-oriented general purpose timesharing system, also based on a DEC PDP-1, which came to be called Zeus. Among its many innovations were the first display-oriented interactive text editor. Because the PDP-1 was not a powerful processor, however, this system was interfaced to a disk on the Computation Center's nearby IBM 7090 so that jobs requiring a lot of crunching could be passed through the disk buffer, run in the batch system there, and returned to the timesharing system for interactive examination of the results.

I joined McCarthy at Stanford in late 1965 and we subsequently put together the Stanford Artificial Intelligence Laboratory (SAIL) in an abandoned laboratory building in the foothills above the Stanford campus, near Felt Lake. The first computer there was a DEC PDP-6, installed in June 1966. After a false start with a contractor who couldn't deliver, a 6-console display system that drew text and vectors with a random-access electron beam was added in 1967. The computer system eventually evolved into a dual-processor DEC-10 and continued to provide display-based timesharing services to the Stanford community until 1992. It used a home-grown timesharing system called WAITS that was similar to TOPS-10 in outline but considerably different in detail.

Some people have claimed that "windows" were invented at Xerox PARC or SRI, but their immediate precursors were the "pieces-of-glass" that were part of the SAIL display system from the beginning. The main difference between pieces-of-glass and windows was that the former were transparent (i.e. you could see the lower layers) whereas "windows" were opaque.

A fancier display system, installed at SAIL in 1971, put a terminal using a television monitor on everyone's desk. SAIL was apparently the first system in the world that put terminals in offices -- before that, the few computer displays that

existed were kept in "display rooms." This display system also included an advanced keyboard that introduced the "Meta" key and other features to facilitate touch-typing. That keyboard design was picked up promptly by MIT and Carnegie-Mellon University and later by Apple, whose Command key is a direct descendent of the Meta key on the SAIL keyboard.

By 1972 the display system included a digital video switch that allowed users to select rapidly from a variety of computer-generated images or other video sources, including commercial television. There was also a speaker on each work station and a novel audio switch that used digital components to allow selection from several audio sources.

The original PDP-6 system had just 64k words of storage (which occupied eight large cabinets) and used microtapes for secondary storage. A fixed-head disc file built by Librascope, added in 1968, was supposed to function both as a swapping store and a permanent file store, but it turned out to be so temperature-sensitive that it was useless for file storage. The six remarkably large discs in this system, which were each 4 feet in diameter, were eventually sold as coffee tables -- I have one in my living room. Despite its large physical size, this disc system had a capacity of only about 100 megabytes. More reliable disks made by IBM, Ampex and DEC were added in later years.

A number of people joined SAIL in the late 1960s, including Don Knuth, who later went off on his own but continued to use the SAIL computer as his main "home" because of its many advanced features. Raj Reddy, who had just finished his Ph. D. at Stanford, continued his pioneering work in speech recognition and eventually moved it to Carnegie-Mellon University.

Another recent Ph. D. named John Chowning developed his ideas on computer synthesis of music at SAIL, leading to a patented synthesizer that was licensed to Yamaha and that made millions of dollars for him and for Stanford. Chowning later formed a computer music research group called CCRMA (Center for Computer Research in Music and Acoustics).

Art Samuel had joined the Lab in 1967 after retiring from IBM. He continued to develop his checkers program, which was the world champion at that time. One of his students developed the most advanced Go program of that era.

Dr. Kenneth Colby joined the Lab in 1968 and his group developed a number of experimental natural-language-understanding programs, including Parry, which answered questions in a manner that simulated the responses of a paranoid person.

Among the user-friendly features of SAIL was an advanced version of Spacewar, a rockets-and-torpedoes game created principally by Steve (Slug) Russell, who had developed the first version while he was at MIT. That idea was further developed by a couple of our staff members into a commercial version using a PDP-11 computer. It became quite popular at a local bowling alley and at the Stanford coffee shop, but the developers knew nothing about how to run a business and their small enterprise went nowhere.

Meanwhile, a guy named Nolan Bushnell picked up the same idea and formed a small company called Atari that developed Spacewar as their first product. Deciding that it was too complicated to be a marketing success, they sold it to another company, and went on to develop a simpler game that turned out to be quite popular; it was called Pong....

A grad student named Don Woods later took a game idea from another person and developed Adventure, which spread over the ARPAnet [predecessor of the Internet] and later evolved in various directions. Today, Adventure is considered the ancestor of almost all text-based computer games.

More serious work on computer gaming included McCarthy's chess program that he had begun at MIT and that was used in a match with one in the Soviet Union. (We lost, but it caused our Russian counterparts a lot of grief when the KGB discovered that we were exchanging telegrams containing what looked like coded messages.)

A DEC consultant named Richard P. Gruen, who used to hang out at SAIL, developed a system for controlling complex program compilations that he called RPG, which theoretically stood for "Rapid Program Generation," but also happened to be his initials. This idea was later incorporated into Unix as the "make" command.

The computer was used for text editing right from the beginning. Bill Weiher and others developed a simple text editor that came to be called SOS and spread throughout the DEC-6/10/20 community. Later a page-oriented editor called E became the primary editor in the Lab. Many features originating with E were incorporated into the emacs editor that was developed later at MIT.

I decided early on that I needed a spelling checker in order to cope with my deficiencies in that area. Fortunately, I happened to have a dictionary of the 10,000 most common English words that I had punched into paper tape when I was at MIT; and during 1960-62, I had developed a spelling checker as a subroutine in a pen-based system for recognizing cursive writing. (This system, which I had also developed, worked at

least as well as the handwriting recognizers that are now appearing on the market.) As I later learned, this 1960 system was evidently the first computer spelling checker developed anywhere.

In 1966 I gave the dictionary to one of our grad students at Stanford, and he wrote a new spelling checker in LISP that clanked a bit but did the job. A few years later, another grad student named Ralph Gorin did a faster one in machine language that included spelling correction. That became quite popular in the lab.

SAIL was connected to ARPAnet around that time, and programs and data began circulating between the research sites through a mixture of donation and benign thievery. Our spelling checker spread to DECsystem-10 and -20 computers all over the net and a Unix version was subsequently developed. Such programs were included later in the personal computers that began appearing in the mid-1970s.

Another program called FINGER, which I developed to help keep track of the unpredictable migrations of our staff at all hours of the day and night, was picked up by several other DEC-10 and DEC-20 computer facilities. We later modified it to work through the ARPAnet and track the denizens of remote computers. It too was rewritten for Unix, but the author of the Unix version was not careful about security, and a loophole in it was exploited much later by the infamous Internet Worm.

Another area enriched by cooperation and innocent larceny was the development of raster graphics printing, initially based on the Xerox XGP and later on laser printers developed by Xerox, Canon and others. Larry Tesler and I had developed an early text formatting program called PUB that facilitated printing on line printers, teletypes, and microfilm, which was later modified by a Carnegie-Mellon student to print on the XGP. People at various sites, principally Carnegie-Mellon, Stanford, and MIT, developed font design software and developed a robust collection of typefaces that migrated all over the network.

Inspired by the deficiencies of PUB, a grad student at Carnegie-Mellon named Brian Reid developed another text formatting program called Scribe. Don Knuth also put one together called TeX, which became a pre-eminent standard for scientific and technical page description, and later developed a fancy font design program called Metafont.

I was a member of the ARPA committee that reviewed the initial technical proposals for ARPAnet, and SAIL became part of the original network when it started in 1969, though we had to defer regular network operation until we got enough memory to hold the rather large amount of communications software that

was required.

Naturally, development work at this level created a need for food at all hours of the day and night, accessible with minimal distraction. Around 1972 we developed SAIL's response to this need, a computer controlled vending machine which sold on credit. Called the Prancing Pony after an inn in Tolkien's _Lord of the Rings_, it still operates in the Computer Science Department at Stanford, though both hardware and software have been updated.

In SAIL's enjoyable work environment, researchers did pioneering work on computer vision, robotics, and automated assembly as well as mathematical theory of computation, theorem proving, and "common sense" reasoning. Hans Moravec's system that guided a robot vehicle, using stereoscopic images from a video camera, did pioneering work on navigation and obstacle avoidance.

Several people moved from SAIL to Xerox PARC when it was formed in the early 1970s, including Alan Kay and Larry Tesler, and took the SAIL culture with them. Others later moved to Lucasfilm to develop the computer technologies supporting "Star Wars" and other elaborate flicks.

Some of our students developed the first interactive CAD system for computer design, called SUDS for "Stanford University Drawing System," and used it to design the Super Foonly, which heavily influenced the DEC KL-10 computer. DEC later used SUDS as their primary design tool for over a decade. They also donated a KL-10 to the Lab.

SUDS was also a key resource in the formation of both Foonly, Inc., a small company (now defunct) that made computers that were DEC-10 compatible, and Valid Logic, a pioneering CAD company. SUDS was also used by Andy Bechtolscheim, a co-founder of Sun Microsystems, to design the first SUN workstation (SUN stood for Stanford University Network). Andy continued to use SUDS to design successive Sun workstations, using the 1967-vintage SAIL displays through 1987.

Other commercial spin-offs from SAIL include:

Vicarm, one of the earliest robotics companies, which made high-performance electric arms and was later purchased by General Electric.

Xidex, which developed and marketed a portable compiler called MainSail.

Imagen, which I founded, and which developed and marketed the earliest desktop publishing systems. The company couldn't get

funding, because the venture capitalists had never heard of laser printers and were not convinced that there was a market for them, but it bootstrapped to annual sales of around $20 million before being purchased by QMS.

Lucid, which developed and marketed LISP compilers and related software.

Cisco, which appropriated Stanford-developed digital communications technology, and eventually got a license from Stanford after being threatened with legal action.

In 1979 SAIL rejoined the computer science department in a new building on the main Stanford campus, but effectively lost its organizational identity in the process. The DEC-10 computer called SAIL continued to operate for another dozen years, providing a comfortable "home" for those who had come to appreciate its features. A party was held on June 7, 1991 to celebrate SAIL's 25th birthday. It was by that time the oldest "living" timesharing system in the world.

However, SAIL was no longer maintained, and began exhibiting the computer equivalent of senile dementia. The computer was powered down for the last time and dismantled on October 4, 1991, but is still fondly remembered by many who used it for work and play over the decades. It was replaced by a small DEC workstation running Unix, also called SAIL, which has much more memory and happens to be much faster than the old SAIL computer; but it has much less character.

THE DISCOLOURATION OF PLASTIC COMPUTER CASES

----

by Dr. Edward Then
Senior Conservator (New Materials)
Science Museum
LONDON SW7 2DD, United Kingdom

The yellowing or discolouration of computer cases is an extremely common phenomenon. The problem is not unique to cases made by one manufacturer, nor is it restricted to computer casings. This chemical process is comparable to the discolouration of an apple skin, and is similarly irreversible. Fortunately, in most instances the damage associated with discolouring affects only the surface of the artifact.

Background: POLYMER AGING

The polymer most commonly used in casings and housings for electrical equipment and computers is ABS. The acronym is derived from the initial letters of the three main monomers

used for its manufacture -- acrylonitrile, butadiene and styrene. ABS polymer was first made available in the early 1950s and, since then, has become one of the most widely used industrial polymers. It is valued by producers for its excellent mechanical properties (impact resistance, stiffness, surface quality), thermal properties (good dimensional stability at high temperature) and electrical resistance. It also offers significant resistance to chemical and stress cracking.

Polymers, including ABS, can be described as large molecules made up of simple repeating units; the word _polymer_ is derived from the Greek words 'poly' and 'mer' meaning "many" and "part" respectively. Many types of polymers can be created by varying the molecular composition of the repeating unit. The total number of repeat units in a polymer chain, often referred to as the _degree of polymerisation_, may typically be hundreds or more.

During degradation, different chemical reactions occur along the polymer chain. These can result in the breaking and rearranging of chemical bonds, causing (among other things!) discolouration of the polymer. Degradation may be initiated or accelerated by numerous factors including ultraviolet light (UV), visible light, ozone and other extraneous pollutants, intrinsic manufacturing impurities, oxygen, and heat. In the case of our computer housing, I think UV and light are the main causes of deterioration. The rate of deterioration is thought to be approximately proportional to the light intensity.

Deterioration-fighting chemicals are commonly added to polymers during manufacturing; these may include antioxidants, antiozonations, light stabilisers, UV stabilisers and fire retardants. The type of additive used will be determined by the composition and application of the finished product. As the polymer ages, most of these additives are consumed while they hold back the degradation process; once the stabilisers are used up, the polymer is often left unprotected and will deteriorate very rapidly. Attempts have been made to restabilise polymers, but it is not known how well these will work, and the topic demands considerable exploration. The Science Museum in collaboration with other institutions is currently sponsoring research in this area.

WHAT SHOULD WE DO NOW, AND HOW CAN WE EXTEND THE LIFE SPAN?

The best advice is, perhaps, to do nothing. Personally I would advise that discoloured surfaces should be left untreated. Maybe, one day, the discolouration will be seen as desirable or inevitable, like the patina on metals! In any case, each example must be evaluated individually, preferably by a

conservator who deals with plastics.

Dirt and grime are a separate problem, and may be cleaned with distilled or deionised water. Stubborn stains can be removed with a non-ionic detergent. The cleaned surface must then be dried immediately. A word of CAUTION: When cleaning with water, use a cloth or cotton wool that is only slightly damp, and avoid making contact with metal parts -- which may corrode -- and with the electronics. Avoid using solvent; some solvents may appear harmless on contact, but will react with the plastic over time, crazing or cracking the object later. Use only soft cloth or cotton wool to dry the case to avoid abrasion or scratching.

Until there is a solution to this problem, the only prudent strategy is preventive conservation. Try to keep the computer away from strong light, especially direct sunlight and other strong UV, and from any heat source. Also keep it covered when it is not being used, to forestall the build-up of dust. I will try to keep the editor informed on the progress of the research projects.

BERKELEY'S UCBVAX CHANGES JOBS

----

by Cliff Frost
Network Services
Computer Science Department, UC Berkeley

On Friday, August 19, 1994, at approximately 2 p. m. Pacific Time, a group of programmers gathered in the old CS department computer room, fourth floor of Evans Hall, UC Berkeley, amongst the scattered remains of network wiring, ancient hardware, ghosts of legends, and general debris, for a mysterious and moving ceremony; a rite of passage for a computer, and perhaps for its human caretakers. A semiologist could write a thesis on this event, but here we confine ourselves to the facts. Keith Sklower opened with a brief history of ucbvax:

"In the summer of 1978, the computer science department took delivery of the campus' first Digital Equipment VAX computer, obtained via a grant from NSF (due in large part to the efforts of Prof. Richard Fateman). In fairly short order, it was running a variant of UNIX developed by Bell Labs. (Local CS people were interested in adding virtual memory support, which ATT UNIX lacked, which eventually led to the widespread interest in BSD, but that's a different story).

"The people at Bell Labs offered to have their computer call up ours in order to facilitate research collaborations, using the UUCP protocol. We had to choose a node name, and 'ucbvax' seemed to follow their naming conventions. Netnews and the

birth of USENET followed closely.

"After about three years, the mail-handling and news functions of the departmental VAX were chewing up more than half the cycles, so it was decided to segregate those functions onto a separate machine, when the opportunity arose. So 'ucbvax' became a VAX/750 devoted specifically to those functions. For a long time, it was one of two gateways between the ARPAnet and the Berkeley campus.

"As the load on ucbvax-the-750 began to exhaust its capacity, there was talk of replacing it with some flavor of SUN workstation, but a DEC sales person got wind of this and thought it would be much better for DEC if ucbvax were to stay a VAX, and managed to 'upgrade' the 750 to a DECstation 3200, its final incarnation."

Keith then assured the assembled company that although the machine was retiring, it would be following recent local tradition by immediately coming back to work in another capacity -- as a card-key access system controller for the UC Police department. His eulogy was followed by Eric Allman's moving tribute:

"Alas, poor ucbvax! I knew him, Horatio. A machine of infinite jest, of most excellent software. He hath borne my mail in his queue a thousand times. And now how abhorred in my imagination it is! My gorge rises at it. Here hung those disks that have spun I know not how oft. Where be your news now? your dialins? your routes? your flashes of congestion that were wont to set the department on a roar?"

The actual power-down was delayed by about 20 minutes, as about 74 pieces of queued email were moved to another machine for eventual processing. Following this, Keith and Eric halted the computer that had carried the name of "ucbvax.berkeley.edu" for the last several years of that venerable name's history. Kirk McKusick turned the power off, an honor due to him as the first person to power-on that particular piece of hardware. At the time it was turned off, ucbvax was the last operational VAX in CS, so -- to stretch the truth only a little -- it was both the first and last VAX in the Computer Science department at UC Berkeley.

The assembled multitude of T-shirted and blue-jeaned programmers (more than a few of us also sporting just a touch of silver in the hair) applauded enthusiastically, then dug into carrot cake and diet Pepsi generously provided by Keith.

Although ucbvax's IP addresses are retired, a significant amount of email traffic is still being supported under that name, rerouted earlier (through the magic of MX records) to a

machine supported by Information Systems and Technology's department of Data Communication and Network Services. This computer is named, with full cognizance of the irony, "nak.berkeley.edu".